IN THE CLAIMS:

Please amend claims as indicated in the complete listing of claims provided below.

1. (Currently Amended) A method for execution by a microprocessor in response to receiving a single instruction, the method comprising:

receiving a first plurality of numbers and a second plurality of numbers; and

generating a third plurality of numbers, each of which is an absolute difference between a number in the first plurality of numbers and a number in the second plurality of numbers;

wherein the third plurality of numbers are saved in an entry in a register file;

wherein the above operations are performed in response to the microprocessor receiving the single instruction.

2. (Original) A method as in claim 1 wherein an absolute difference between a first number and a second number is computed using a method comprising:

producing a first intermediate number by subtracting the second number from the first number;

producing a second intermediate number by subtracting the first number from the second number; and

selecting a positive number from the first intermediate number and the second intermediate number as the absolute difference between the first number and the second number;

wherein the microprocessor is a media processor disposed on an integrated circuit with a memory controller.

3. (Original) A method as in claim 2 wherein the first intermediate number and the second intermediate number are produced in parallel; and wherein the third plurality of numbers are generated substantially simultaneously.

4. (Currently Amended) A method as in ~~claim 3~~ claim 2 further comprising:

testing if an overflow occurs in producing the first intermediate number and the second intermediate number;

saturating the absolute difference between the first number and the second number if an overflow occurs.

5. (Currently Amended) A method as in claim 1 wherein the first plurality of numbers are received from ~~an entry in a~~ a first entry in the register file.

6. (Currently Amended) A method as in claim 5 wherein the single instruction specifies a way to partition a string of bits in the first entry into the first plurality of numbers.

7. (Currently Amended) A method as in claim 5 wherein the single instruction specifies an index of the first entry in the register file.

8. (Canceled)

9. (Currently Amended) A method as in ~~claim 8~~ claim 1 wherein the single instruction specifies an index of the entry in the register file.

10. (Original) A method as in claim 1 wherein a type of each of the first and second pluralities of numbers is one of:

a) unsigned integer;

b) signed integer; and

c) floating point number.

11. (Original) A method as in claim 1 wherein a size of each of the first and second pluralities of numbers is one of:

a) 8 bits;

b) 16 bits; and

c) 32 bits.

12. (Currently Amended) A machine readable media containing an executable computer program instruction which when executed by a digital processing system causes said system to perform a method comprising:

receiving a first plurality of numbers and a second plurality of numbers; and

generating a third plurality of numbers, each of which is an absolute difference between a number in the first plurality of numbers and a number in the second plurality of numbers;

wherein the third plurality of numbers are saved in an entry in a register file;

wherein the above operations are performed in response to a microprocessor of the digital processing system receiving the instruction.

13. (Original) A media as in claim 12 wherein an absolute difference between a first number and a second number is computed using a method comprising:

producing a first intermediate number by subtracting the second number from the first number;

producing a second intermediate number by subtracting the first number from the second number;

selecting a positive number from the first intermediate number and the second intermediate number as the absolute difference between the first number and the second number.

14. (Original) A media as in claim 13 wherein the first intermediate number and the second intermediate number are produced in parallel.

15. (Currently Amended) A media as in ~~claim 14~~ claim 12 wherein the method further comprises:

testing if an overflow occurs in producing the first intermediate number and the second intermediate number;

saturating the absolute difference between the first number and the second number if an overflow occurs.

16. (Currently Amended) A media as in claim 12 wherein the first plurality of numbers are received from ~~an entry in a~~ a first entry in the register file.

17. (Currently Amended) A media as in claim 16 wherein the ~~single~~ instruction specifies a way to partition a string of bits in the first entry into the first plurality of numbers.

18. (Currently Amended) A media as in claim 16 wherein the ~~single~~ instruction specifies an index of the first entry in the register file.

19. (Canceled)

20. (Currently Amended) A media as in ~~claim 17~~ claim 12 wherein the ~~single~~ instruction specifies an index of the entry in the register file.

21. (Original) A media as in claim 12 wherein a type of each of the first and second pluralities of numbers is one of:

a) unsigned integer;

b) signed integer; and

c) floating point number.

22. (Original) A media as in claim 12 wherein a size of each of the first and second pluralities of numbers is one of:

a) 8 bits;

b) 16 bits; and

c) 32 bits.

23. (Currently Amended) An execution unit in a microprocessor, the execution unit comprising:

a first circuit configured to receive a first plurality of numbers;

a second circuit configured to receive a second plurality of numbers; and

a third circuit coupled to the first circuit and the second circuit, the third circuit, in response to the microprocessor receiving a single instruction, generating a third plurality of numbers, each of which is an absolute difference between a number in the first plurality of numbers and a number in the second plurality of numbers

wherein the microprocessor is a media processor disposed on an integrated circuit with a memory controller.

24. (Original) An execution unit as in claim 23 wherein the third circuit comprises a plurality of units, each of the plurality of units comprising:

a fourth circuit configured to generate a first intermediate number by subtracting a number in the second plurality of numbers from a number in the first plurality of numbers;

a fifth circuit configured to generate a second intermediate number by subtracting a number in the first plurality of numbers from a number in the second plurality of numbers; and

a sixth circuit coupled to the fourth circuit and the fifth circuit, the sixth circuit selecting a positive number from the first intermediate number and the second intermediate number as an absolute difference between the first number and the second number.

25. (Original) A processing system comprising an execution unit as in claim 23.

26. (Currently Amended) An execution unit in a microprocessor, the execution unit comprising:

means for receiving a first plurality of numbers and a second plurality of numbers; and

means for generating a third plurality of numbers, each of which is an absolute difference between a number in the first plurality of numbers and a number in the second plurality of numbers;

wherein the third plurality of numbers are saved in an entry in a register file;

wherein the above means perform operations in response to the microprocessor receiving ~~the~~ a single instruction.

27. (Original) An execution unit as in claim 26 wherein an absolute difference between a first number and a second number is computed using a unit comprising:

means for producing a first intermediate number by subtracting the second number from the first number;

means for producing a second intermediate number by subtracting the first number from the second number;

means for selecting a positive number from the first intermediate number and the second intermediate number as the absolute difference between the first number and the second number.

28. (Original) An execution unit as in claim 27 wherein the first intermediate number and the second intermediate number are produced in parallel.

29. (Currently Amended) An execution unit as in ~~claim 28~~ claim 27 further comprising:

means for testing if an overflow occurs in producing the first intermediate number and the second intermediate number;

means for saturating the absolute difference between the first number and the second number if an overflow occurs.

30. (Currently Amended) An execution unit as in claim 26 wherein the first plurality of numbers are received from ~~an entry in a~~ a first entry in the register file.

31. (Currently Amended) An execution unit as in claim 30 wherein the single instruction specifies a way to partition a string of bits in the first entry into the first plurality of numbers.

32. (Currently Amended) An execution unit as in claim 30 wherein the single instruction specifies an index of the first entry in the register file.

33. (Canceled)

34. (Currently Amended) An execution unit as in ~~claim 33~~ claim 26 wherein the single instruction specifies an index of the entry in the register file.

35. (Original) An execution unit as in claim 26 wherein a type of each of the first and second pluralities of numbers is one of:

a) unsigned integer;

b) signed integer; and

c) floating point number.

36. (Original) An execution unit as in claim 26 wherein a size of each of the first and second pluralities of numbers is one of:

a) 8 bits;

b) 16 bits; and

c) 32 bits.

37. (new) A method as in claim 1, wherein a type of each of the first and second pluralities of numbers is floating point number.

38. (new) A media as in claim 12, wherein the microprocessor is a media processor disposed on an integrated circuit with a memory controller.

39. (new) An execution unit as in claim 26 further comprising:

means for testing if an overflow occurs.